Generative Bayesian Models For Discrete Data

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

January 18, 2019

References

- Kevin Murphy, Machine Learning: A probabilistic perspective, Chapter 3
- C P Robert, The Bayesian Choice: From Decision-Theoretic Motivations to Compulational Implementation, Springer-Verlag, NY, 2001 (online resource)
- A Gelman, JB Carlin, HS Stern and DB Rubin, Bayesian Data Analysis, Chapman and Hall CRC Press, 2nd Edition, 2003.
- J M Marin and C P Robert, The Bayesian Core, Spring Verlag, 2007 (online resource)
- D. Sivia and J Skilling, Data Analysis: A Bayesian Tutorial, Oxford University Press, 2006.
- Bayesian Statistics for Engineering, Online Course at Georgia Tech, B. Vidakovic.
- Additional References with links are provided in the lecture slides.

Contents

- Introduction: Generative Models
- Bayesian concept learning, Likelihood, Prior, Posterior, Posterior predictive distribution, Plug-in Approximation, A more complex prior
- The beta-binomial model, Likelihood, Prior, Posterior, Posterior predictive distribution, Blackswan paradoxes and Plug-in approximations, Outcome of multiple future trials, Beta-Binomial Distribution
- The Dirichlet-multinomial model, Likelihood, Prior, Posterior, Posterior predictive, Bayesian Analysis of the Uniform Distribution, Language Model using Bag of Words
- Naive Bayes classifiers, Examples, MLE for Naïve Mayes Classifier, Example for bag-of-words 2 class model, Summary of the Algorithm, Bayesian Naïve Bayes, Using the model for prediction, The log-sum-exp trick, Feature selection using mutual information
- Classifying documents using bag of words

Introduction: Generative Models

- We discuss how to classify a feature vector $\boldsymbol{x}$ by applying Bayes rule to a generative classifier of the form:

$$p(y=c \mid \mathbf{x}, \boldsymbol{\theta}) \propto p(\mathbf{x} \mid y=c, \boldsymbol{\theta})\, p(y=c \mid \boldsymbol{\theta})$$

- The key to using such models is specifying a suitable form for the *class-conditional density* $p(\boldsymbol{x}|y=c,\boldsymbol{\theta})$, *which defines what kind of data we expect to see in each class.*

- We focus on the case where the observed *data are discrete*.

- We also discuss how to infer the unknown parameters $\boldsymbol{\theta}$ of such models.

Bayesian Concept Learning

- Concept learning is equivalent to binary classification, e.g. define $f(x) = 1$ if x is an example of the concept C, and $f(x) = 0$, otherwise.

- The goal is to learn the indicator function f which defines which elements are in the set C.

- Standard binary classification techniques require positive and negative examples. However, here we consider a way to *learn only from positive examples.*

- We will consider a simple example of concept learning called the *number game*.

- Tenenbaum, J. (1999). *A Bayesian framework for concept learning*. Ph.D. thesis, MIT.

Bayesian Concept Learning: Number Game

- *Choose a simple arithmetical concept* C, such as "prime number" or "a number between 1 and 10".

- We then give you a series of randomly chosen positive examples $\mathcal{D} = \{x_1, \ldots, x_N\}$ *drawn from* C.

- We now ask you whether some new test case $\tilde{x}$ belongs to C (i.e., we ask you to classify $\tilde{x}$).

- Tenenbaum, J. (1999). *A Bayesian framework for concept learning*. Ph.D. thesis, MIT.

Bayesian Concept Learning

- Given: all numbers are integers between 1 and 100.
- Given: '16' is a positive example of the concept.
- Question: What other numbers do you think are positive?
  17? 6? 32? 99?
- Analysis: It is hard to tell with only one example, so your predictions will be vague. Numbers that are similar in "some sense" to 16 are likely.
- Similar in what way?
  - ✓ 17 is similar, because it is "close by",
  - ✓ 6 is similar because it has a digit in common,
  - ✓ 32 is similar because it is also even and a power of 2,
  - ✓ but 99 does not seem similar.

Posterior Predictive Distribution

- Analysis: We can represent this as a probability distribution, $p(\tilde{x}|D)$ which is the probability that $\tilde{x} \in C$ given the data $\mathcal{D}$ for any $\tilde{x} \in \{1,2,\ldots,100\}$
- This is called the *posterior predictive distribution*.

Empirical Predictive Distribution

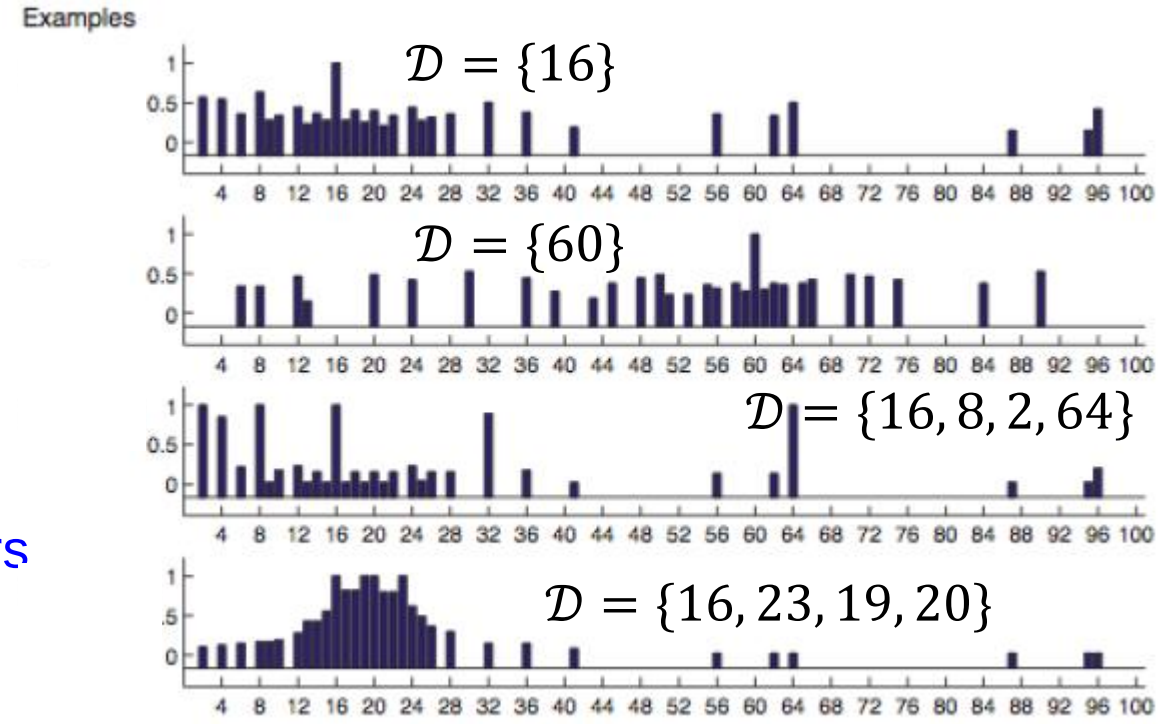
The empirical predictive distribution averaged over 8 persons in the number game is shown. E.g.

- *Given:* $\mathcal{D} = \{16, 8, 2\ 64\}$

  Analysis: guess that the hidden concept is "powers of two" (an example of induction)

- *Given*: $\mathcal{D} = \{16,23,19,20\}$

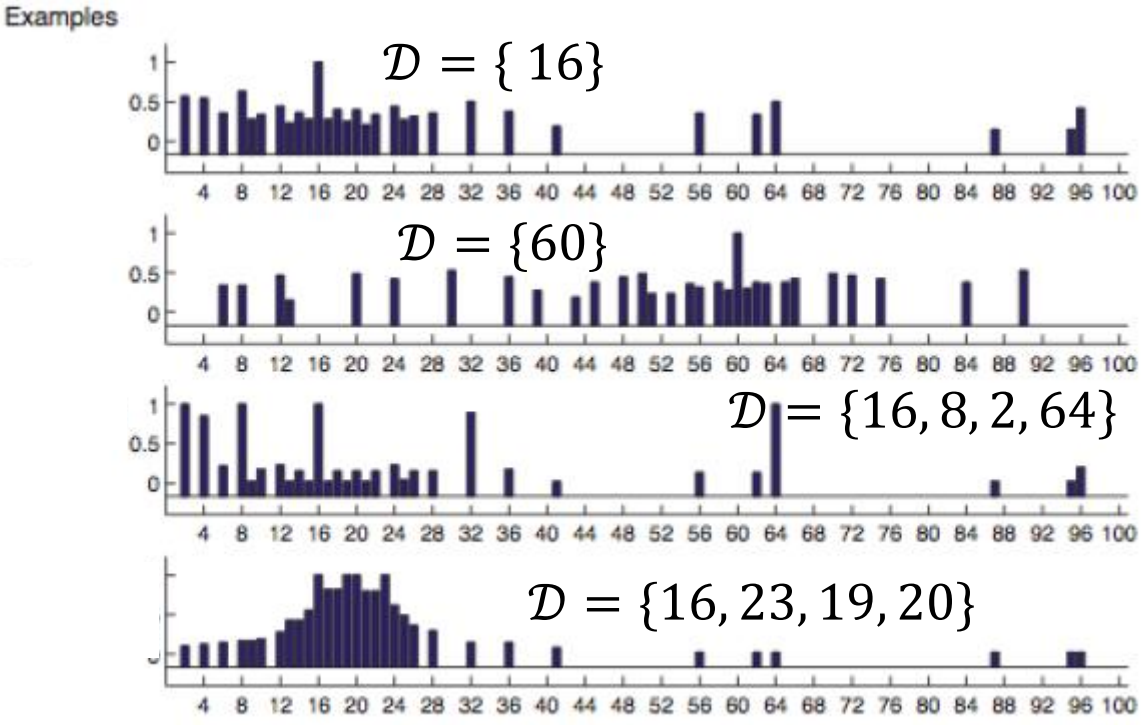
  Analysis: you will get a different kind of generalization (numbers near 20)



- Tenenbaum, J. (1999). *A Bayesian framework for concept learning*. Ph.D. thesis, MIT.

Hypothesis Space of Concepts

- The classic approach to induction is to assume that we have a *hypothesis space of concepts, H*, such as:
  - ✓ odd numbers,
  - ✓ even numbers,
  - ✓ all numbers between 1 and 100,
  - ✓ powers of two, all numbers ending in j (for $0 \leq j \leq 9$),
  - ✓ etc.



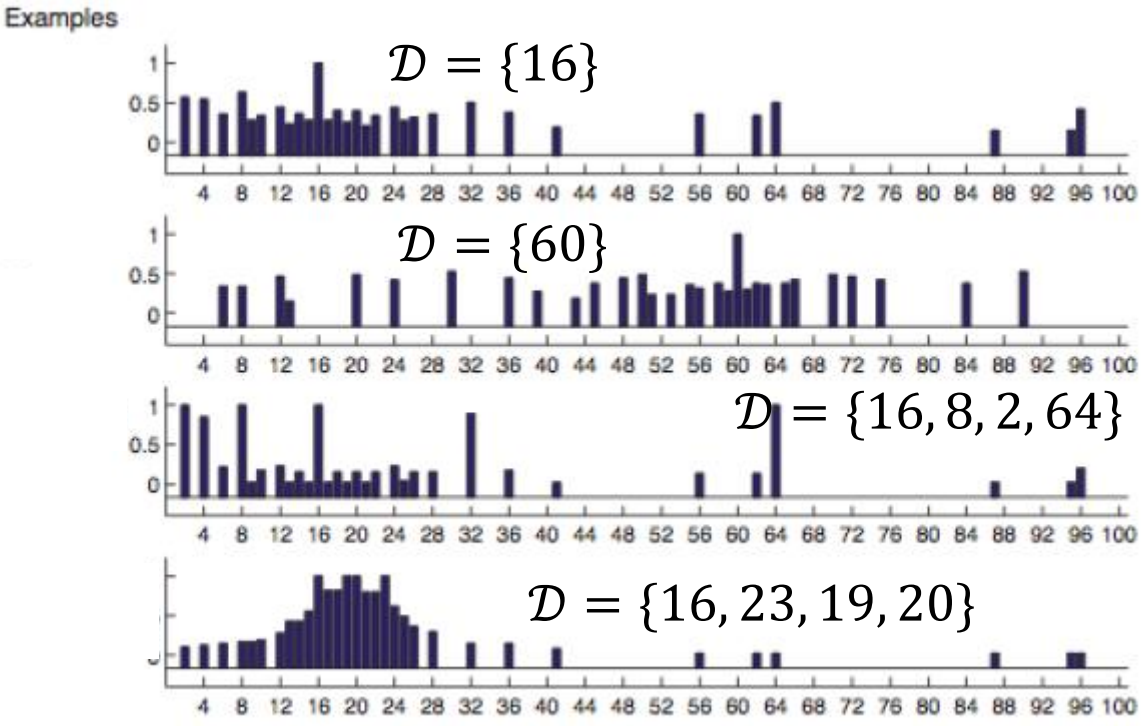
Empirical Predictive Distribution

- Tenenbaum, J. (1999). *A Bayesian framework for concept learning*. Ph.D. thesis, MIT.

The Version Space

- The classic approach to induction is to assume that we have a *hypothesis space of concepts,* H, such as:

- Tenenbaum, J. (1999). *A Bayesian framework for concept learning*. Ph.D. thesis, MIT.



Empirical Predictive Distribution

- The subset of H that is consistent with the data $\mathcal{D}$ is *called the version space*.

- As we see more examples, the version space shrinks and we become increasingly certain about the concept.

Likelihood

- Why we choose h_{two} ="powers of two", and not, say, h_{even} = “even numbers”, after seeing $\mathcal{D} = \{16,8,2,64\}$? Can we provide a Bayesian explanation of this.

- We assume that examples are sampled uniformly at random from a concept h.

- Given this assumption, the probability of independently sampling N items (with replacement) from h is given by

$$p(\mathcal{D} \mid h) = \left(\frac{1}{size(h)} \right)^N = \left(\frac{1}{|h|} \right)^N$$

- This crucial equation embodies what Tenenbaum calls *the size principle*, which means *the model favors the simplest (smallest) hypothesis consistent with the data.*

  - Tenenbaum, J. (1999). *A Bayesian framework for concept learning*. Ph.D. thesis, MIT.
  - Tenenbaum, J. and F. Xu (2000). Word Learning as Bayesian Inference. In Proc. 22nd Annual Conf. of the Cognitive Science Society.

Likelihood

- Example: let $\mathcal{D} = \{ 16 \}$. Then $p(\mathcal{D}|h_{two}) = 1/6$, since there are only 6 powers of two less than 100, but $p(\mathcal{D}|h_{even}) = 1/50$, since there are 50 even numbers. So the likelihood that $h = h_{two}$ is higher than if $h = h_{even}$.

- After 4 examples, the likelihood of $h = h_{two}$ is $(1/6)^4 = 7.7\ 10^{-4}$, whereas the likelihood of h_{even} is $(1/50)^4 = 1.6\ 10^{-7}$. This is a likelihood ratio of almost $5000{:}\ 1$ in favor of h_{two}.

- This quantifies our earlier intuition that $\mathcal{D} = \{16,8,2,64\}$ would be a very suspicious coincidence if generated by h_{even}

Prior

- Although the subjectivity of the prior is controversial, it is quite useful.
- If for example, you are told the numbers are from some arithmetic rule, then given $1200, 1500, 900$ and 1400, you may think 400 is likely but 1183 is unlikely.
- But if you are told that the numbers are examples of healthy cholesterol levels, you would probably think 400 is unlikely and 1183 is likely.
- Thus we see that the prior is the mechanism by which background knowledge can be brought to bear on a problem. *Without having a prior, rapid learning from small samples sizes is impossible.*

Posterior

- The posterior is simply the likelihood times the prior, normalized. In this context we have

$$p(h \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid h)\, p(h)}{\sum_{h' \in H} p(\mathcal{D}, h')} = \frac{p(h)\, \mathbb{I}(\mathcal{D} \in h) / |h|^N}{\sum_{h' \in H} p(h')\, \mathbb{I}(\mathcal{D} \in h') / |h'|^N}$$

where $\mathbb{I}(\mathcal{D} \in h)$ is 1 iff *all* the data are in the extension of the hypothesis h.

- In general, *when we have enough data, the posterior $p(h|\mathcal{D})$ becomes peaked on a single concept,* namely the MAP estimate, i.e.,

$$p(h \mid \mathcal{D}) = \delta_{h^{MAP}}(h)$$

where $\hat{h}^{MAP} = \underset{h}{\operatorname{argmax}} p(h|D)$ is the posterior mode, and where δ is the Dirac measure defined by

$$\delta_x(A) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

MAP Estimate

- Note that the MAP estimate can be written as

$$\hat{h}^{MAP} = \underset{h}{\text{argmax}} p(\mathcal{D}|h)p(h) = \underset{h}{\text{argmax}} [\log p(\mathcal{D}|h) + \log p(h)]$$

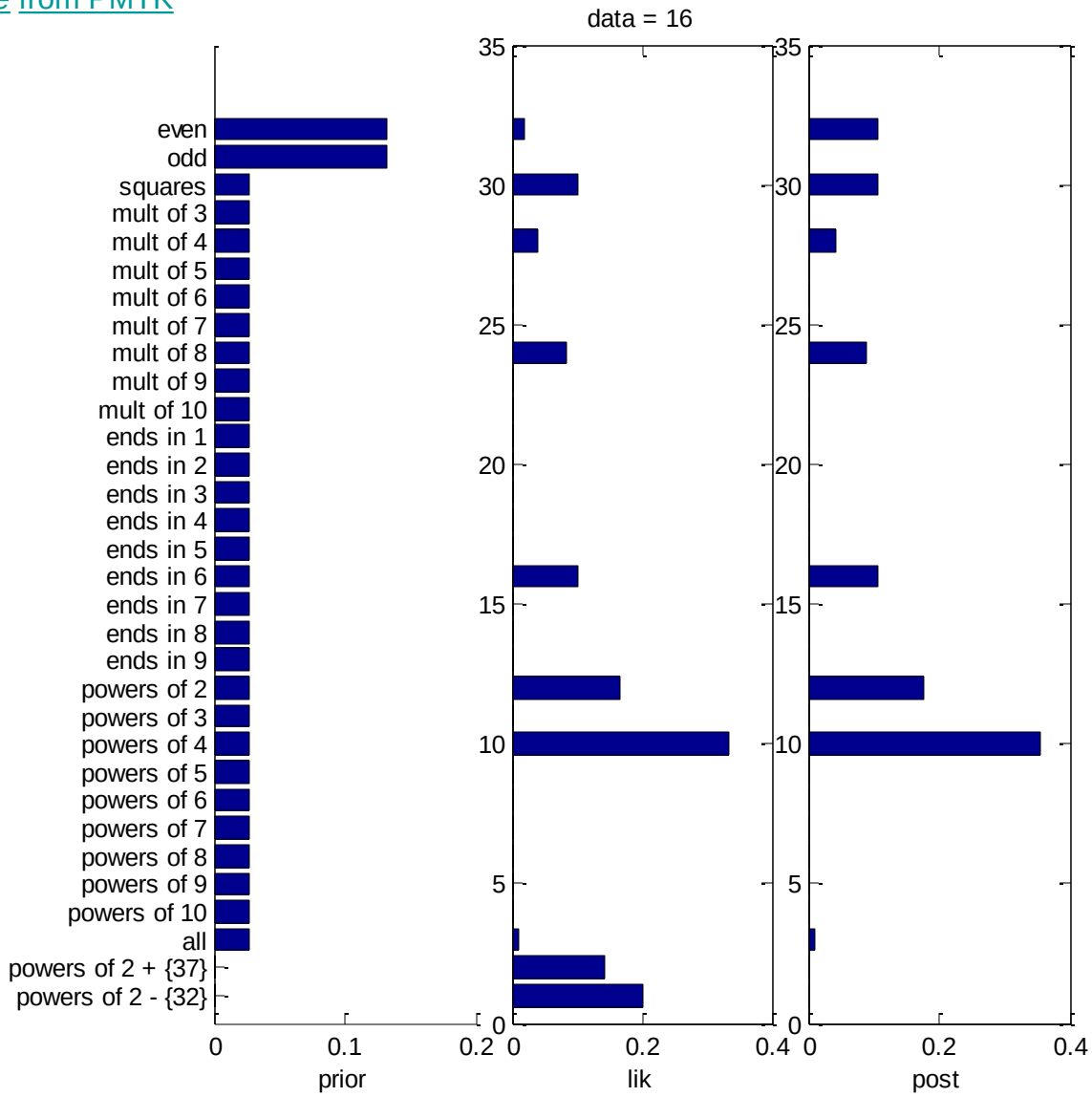
- Since the likelihood term depends exponentially on N, and the prior stays constant, as we get more and more data, the MAP estimate converges towards the maximum likelihood estimate or MLE:

$$\hat{h}^{MLE} = \underset{h}{\text{argmax}} p(\mathcal{D}|h) = \underset{h}{\text{argmax}} [\log p(\mathcal{D}|h)]$$

- *If we have enough data, the data overwhelms the prior. Then, the MAP estimate converges towards the MLE.*

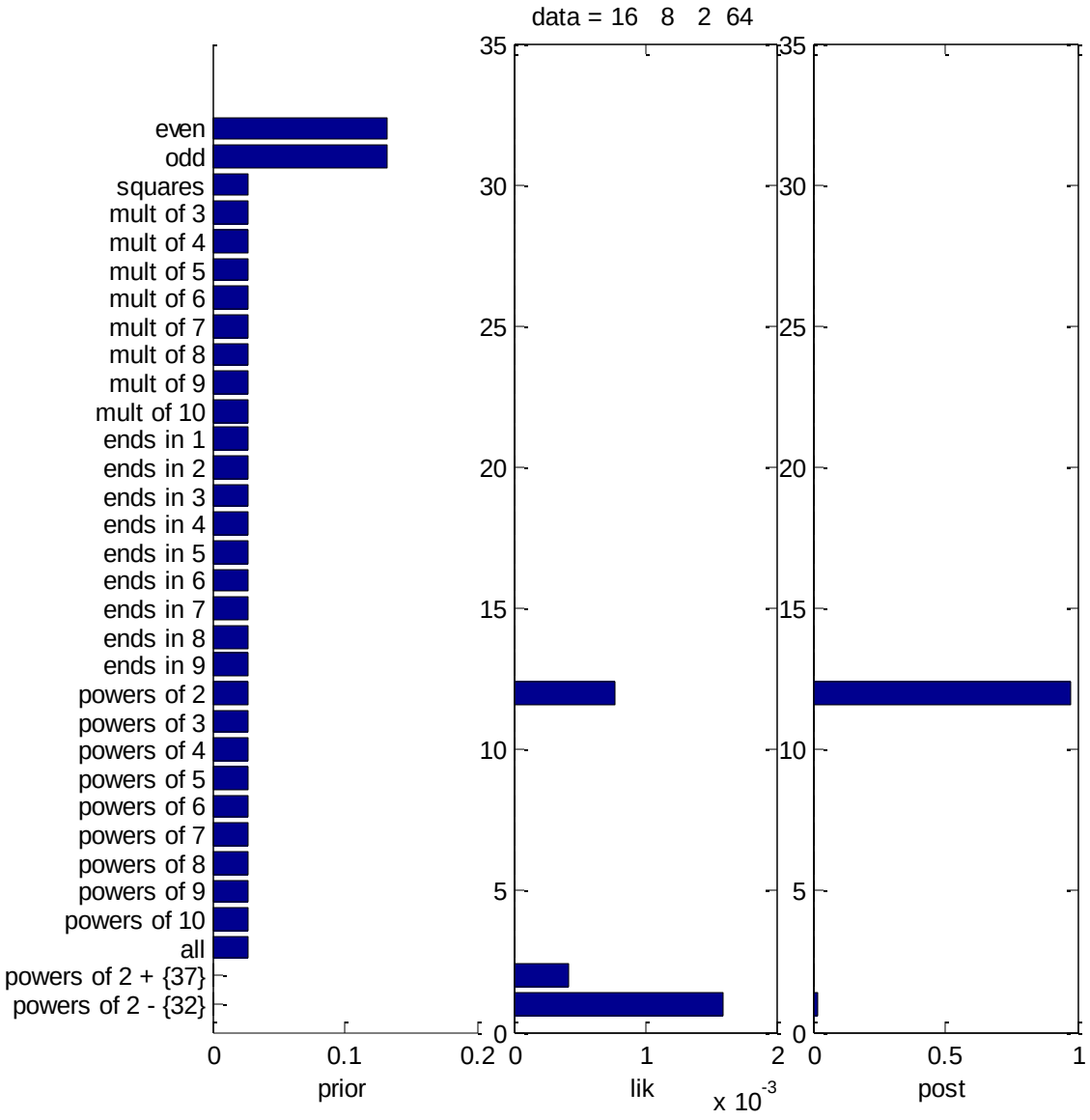
Prior, Likelihood, and Posterior

Run *numbersGame* from PMTK



Prior, likelihood and posterior for $\mathcal{D} = \{16\}$

Prior, Likelihood and Posterior



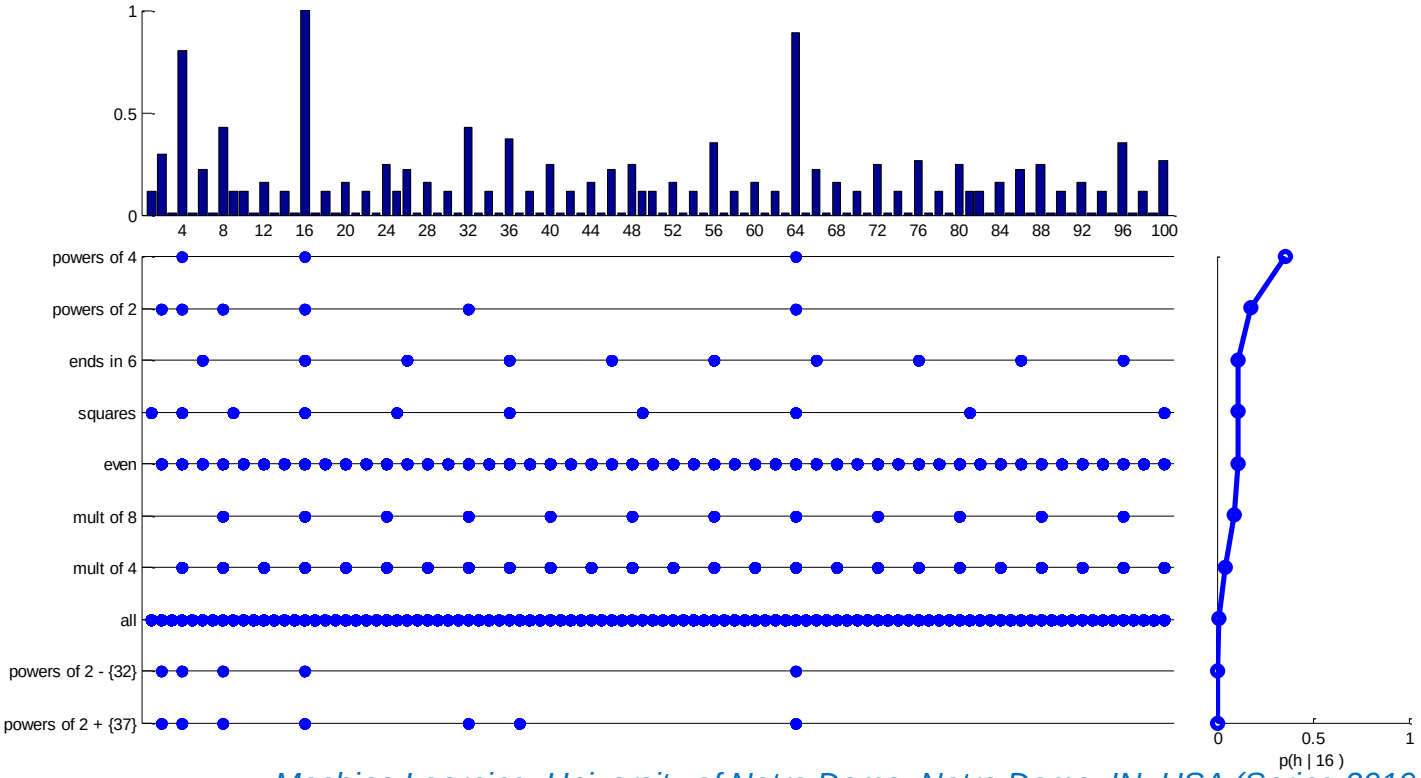
Prior, likelihood and posterior for $\mathcal{D} = \{16,8,2,64\}$

Posterior Predictive Distribution

- The posterior predictive distribution in this context is given by

$$p(\tilde{x} \in C|\mathcal{D}) = \sum_h p(y = 1|\tilde{x}, h)p(h|\mathcal{D})$$

- This is a weighted average of the predictions of each individual hypothesis (*Bayes model averaging*).



- Posterior over hypotheses and the corresponding predictive distribution after seeing $\mathcal{D} = \{16\}$.

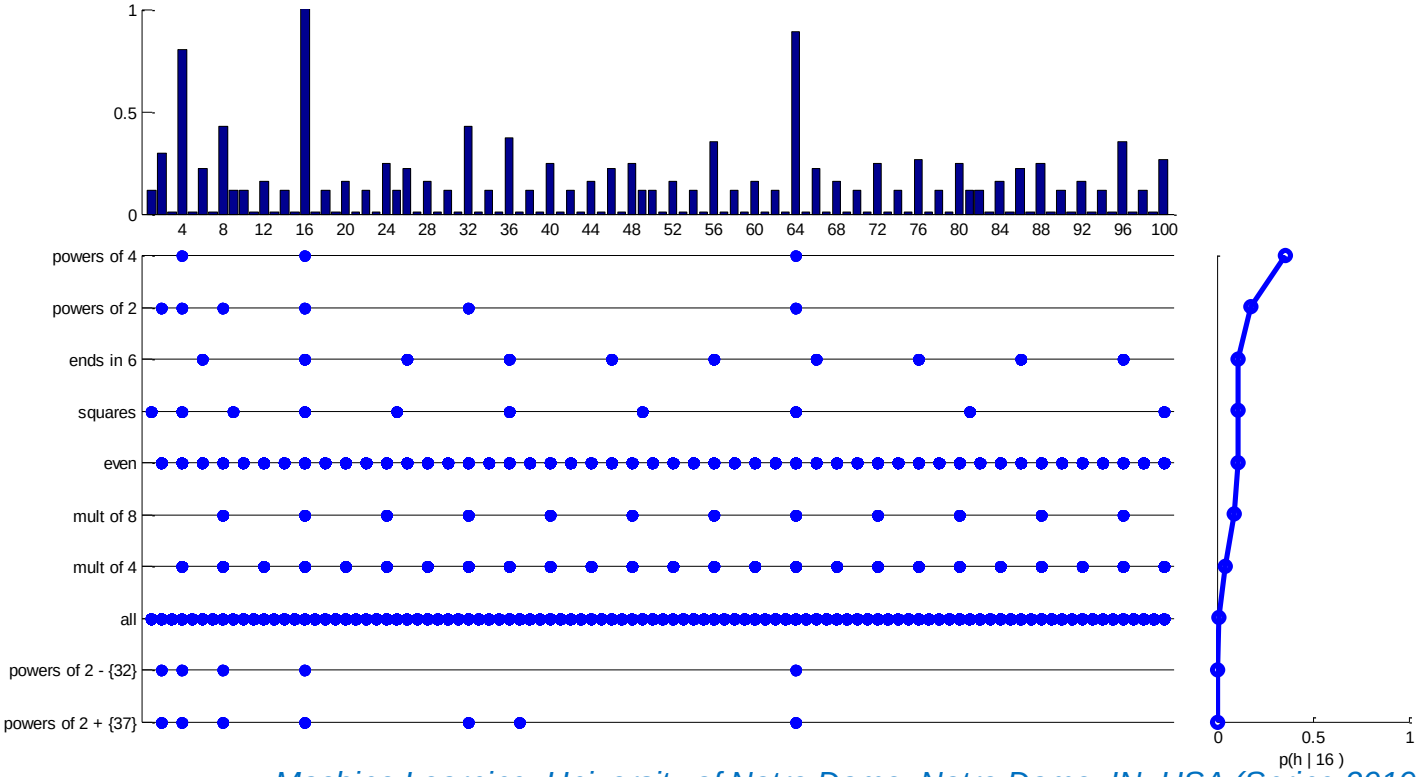
- A dot means this number is consistent with this hypothesis.

Posterior Predictive Distribution

- The posterior predictive distribution in this context is given by

$$p(\tilde{x} \in C|\mathcal{D}) = \sum_h p(y = 1|\tilde{x}, h)p(h|\mathcal{D})$$

- This is a weighted average of the predictions of each individual hypothesis (*Bayes model averaging*).



- The graph $p(h|\mathcal{D})$ on the right is the weight given to hypothesis h.

- By taking a weighed sum of dots, we get:

$$p(\tilde{x} \in C|\mathcal{D})$$

Plug-in Approximation to the Predictive Distribution

- When we have a small dataset, the posterior $p(h|\mathcal{D})$ is vague, which induces a broad predictive distribution.

- However, with a lots of data, the posterior becomes a delta function centered at the MAP estimate. In this case, the predictive distribution is

$$p(\tilde{x} \in C|\mathcal{D}) = \sum_h p(\tilde{x}|h)\delta_{\hat{h}}(h) = p(\tilde{x}|\hat{h})$$

- *This is called a plug-in approximation to the predictive density and is very widely used, due to its simplicity.*

- Hoeting, J., D. Madigan, A. Raftery, and C. Volinsky (1999). Bayesian model averaging: A tutorial. *Statistical Science 4*(4).

Complex Prior: $p(h) = \pi_0 p_{rules}(h) + (1 - \pi_0) p_{\text{interval}}(h)$

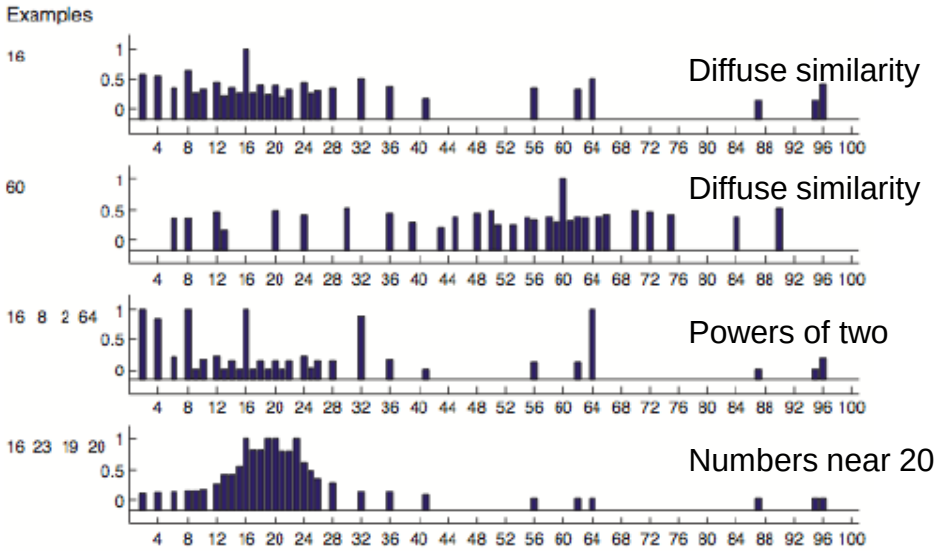
- Let us see a more complex prior.
- To model human behavior, Tenenbaum used a slightly more sophisticated prior which was derived by analyzing some experimental data of how people measure similarity between numbers.
- Thus the prior is a mixture of two priors, one over arithmetical rules, and one over intervals:

$$p(h) = \pi_0 p_{rules}(h) + (1 - \pi_0) p_{\text{interval}}(h)$$

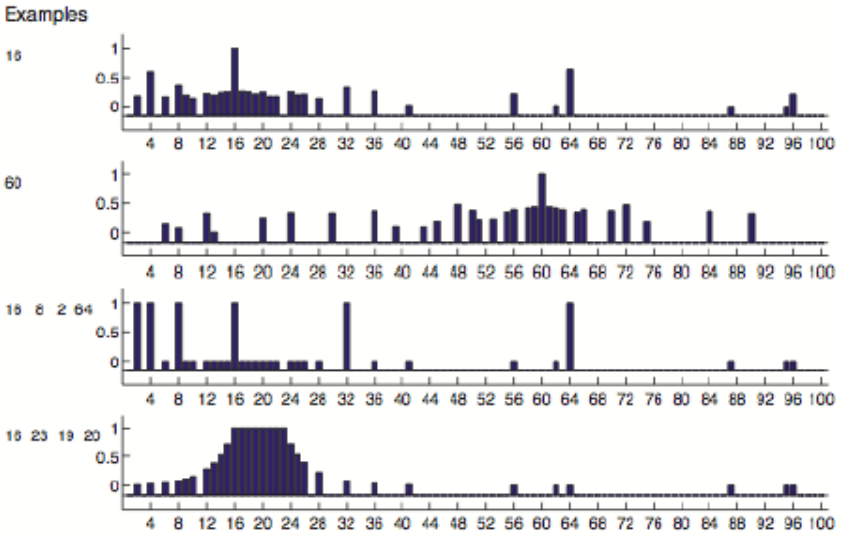
- The results are not that sensitive to π_0 assuming that $\pi_0 > 0.5$

- Tenenbaum, J. (1999). *A Bayesian framework for concept learning*. Ph.D. thesis, MIT.

Complex Prior: $p(h) = \pi_0 p_{rules}(h) + (1 - \pi_0) p_{\text{interval}}(h)$



Empirical predictive distribution averaged over 8 humans in the number game.



Predictive distributions for the model using the full hypothesis space $p(h) = \pi_0 p_{rules}(h) + (1 - \pi_0) p_{\text{interval}}(h)$ (results on are only shown for $\tilde{x}$ for which data is available – compare with the posterior predictive distribution given here)

The Beta-Binomial Model

- We discuss next applications with the unknown *parameters being continuous.*
- Problem of Interest: inferring the probability that a coin shows up heads, given a series of observed coin tosses.
- The coin model forms the basis of many methods including Naive Bayes classifiers, Markov models, etc.
- We specify first the likelihood and prior, and then derive the posterior and predictive distributions.

The Beta-Binomial Model: Likelihood

- Suppose $X_i \sim \mathcal{B}er(\theta)$, where $x_i = 1$ represents "heads", $x_i = 0$ represents "tails", and $\theta \in [0,1]$ is the probability of heads. If the data are iid, the likelihood is:

$$p(\mathcal{D} \mid \theta) = \theta^{N_1}(1-\theta)^{N_0}, \; N_1 = \sum_{i=1}^{N} \mathbb{I}(x_i = 1), \; N_0 = \sum_{i=1}^{N} \mathbb{I}(x_i = 0)$$

- N_1 is the number of heads and N_0 the number of tails in N trials.

- These two counts are the ***sufficient statistics of the data***. This is all we need to know about $\mathcal{D}$ to infer θ.

- Now suppose the data consists of the count of the number of heads N_1 observed in a *fixed number* of N of trials. In this case, we have $N_1 \sim \mathcal{B}in(N, \theta)$, where $\mathcal{B}in$ represents the binomial distribution:

$$\mathcal{B}(k \mid n, \theta) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

The Beta-Binomial Model: Likelihood

$$\mathcal{B}(k \mid n, \theta) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

- Since $\binom{n}{k}$ is a constant independent of θ, the likelihood for the binomial sampling model is the same as the likelihood for the Bernoulli model.

$$p(\mathcal{D} \mid \theta) = \theta^{N_1}(1-\theta)^{N_0},\ N_1 = \sum_{i=1}^{N} \mathbb{I}(x_i = 1),\ N_0 = \sum_{i=1}^{N} \mathbb{I}(x_i = 0)$$

- So *any inference we make about θ will be the same whether we observe the counts, $\mathcal{D} = (N_1, N)$, or a sequence of trials, $\mathcal{D} = (x_1, \ldots, x_N)$.*

Conjugate Prior

- We need a prior with support $[0,1]$. It would be convenient if the prior had the same form as the likelihood, i.e., if the prior looked like

$$p(\theta) \propto \theta^{\gamma_1}(1-\theta)^{\gamma_2}, \gamma_1, \gamma_2 \text{ constants}$$

- Then we could easily evaluate the posterior by simply adding up the exponents:

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)\, p(\theta) = \theta^{N_1+\gamma_1}(1-\theta)^{N_0+\gamma_0}$$

- When the prior and the posterior have the same form, we say that *the prior is a conjugate prior for the corresponding likelihood*. Conjugate priors are widely used since they simplify computation and are easy to interpret.

- In the case of the Bernoulli, the conjugate prior is the Beta distribution.

$$\mathcal{Beta}(\theta \mid a, b) \propto \theta^{a-1}(1-\theta)^{b-1}$$

- *The parameters of the prior are called hyper-parameters*.

Posterior

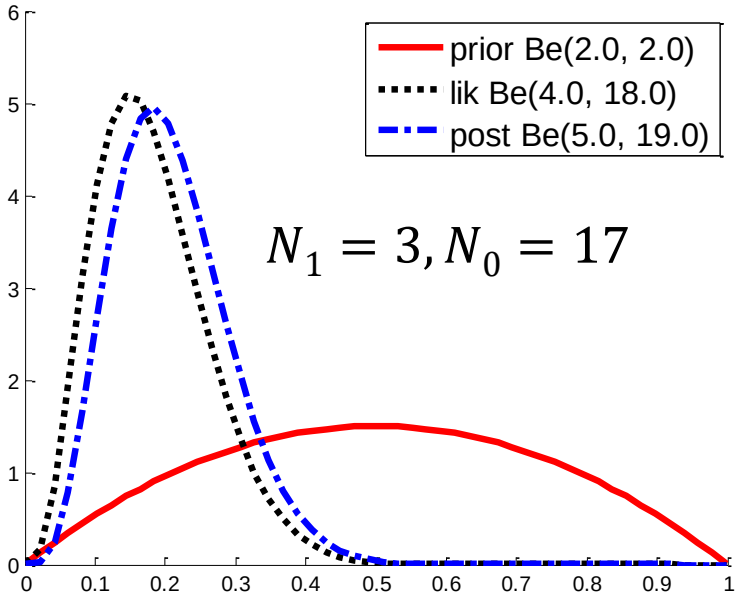
- If we multiply the likelihood by the Beta prior we get the following posterior

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)\, p(\theta) = \theta^{N_1+a-1}(1-\theta)^{N_0+b-1} = \mathcal{Beta}\left(N_1 + a, N_0 + b\right)$$

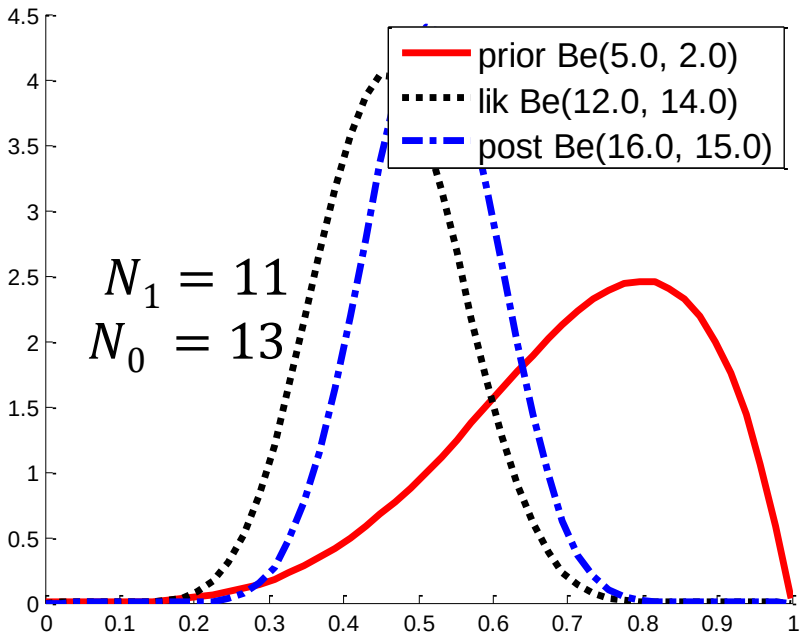
- Note that the posterior is obtained by adding the prior hyper-parameters to the empirical counts.
  - ✓ For this reason, *the hyper-parameters are known as pseudo-counts.*
- *The strength of the prior*, also known as the effective sample size of the prior, *is the sum of the pseudo counts,* $a + b$; this plays a role analogous to the data set size N.

The Beta-Binomial Model: Posterior

- Left figure: we update a weak $\mathcal{B}eta(2,2)$ prior with a *peaked likelihood function*, corresponding to a large sample size; *the posterior is essentially identical to the likelihood* since the data has overwhelmed the prior.
- Right figure: we update *a strong $\mathcal{B}eta(5,2)$ prior* with a peaked likelihood function; now we see that the posterior is a *compromise between the prior and the likelihood*.



Run *binomialBetaPosteriorDemo* from PMTK



Posterior Mean and Mode

$$p(\theta \mid \mathcal{D}) = \mathcal{Beta}\left(N_1 + a, N_0 + b\right)$$

- *The MAP estimate* is given by

$$\hat{\theta}_{MAP} = \frac{N_1 + a - 1}{a + b + N - 2}$$

- *If we use a uniform prior ($a = b = 1$), then the MAP estimate reduces to the MLE*, which is just the empirical fraction of heads:

$$\hat{\theta}_{MLE} = \frac{N_1}{N}$$

- *Let $a_0 = a + b$ be the equivalent sample size of the prior*, and let the prior mean be $m_1 = a / \alpha_0$.

- Then the posterior mean is given by

$$\mathbb{E}[\theta] = \frac{N_1 + \alpha_0 m_1}{\alpha_0 + N} = \frac{\alpha_0}{\alpha_0 + N} m_1 + \frac{N}{\alpha_0 + N} \frac{N_1}{N} = \lambda m_1 + (1 - \lambda)\hat{\theta}_{MLE}$$

$$\theta \sim \text{Beta}(\alpha, \beta)$$
$$p(\theta) = \text{Beta}(\theta|\alpha, \beta)$$

'prior sample sizes'
$$\alpha > 0, \beta > 0$$

$$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}$$
$$\theta \in [0, 1]$$

$$\text{E}(\theta) = \frac{\alpha}{\alpha+\beta}$$
$$\text{var}(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$
$$\text{mode}(\theta) = \frac{\alpha-1}{\alpha+\beta-2}$$

Posterior Mean and Mode

$$\mathbb{E}[\theta] = \frac{N_1 + \alpha_0 m_1}{\alpha_0 + N} = \frac{\alpha_0}{\alpha_0 + N} m_1 + \frac{N_1}{\alpha_0 + N} \frac{N_1}{N} = \lambda m_1 + (1 - \lambda)\hat{\theta}_{MLE}$$

- λ is the ratio of the prior to posterior equivalent sample size.

$$\lambda = \frac{\alpha_0}{\alpha_0 + N}$$

- So the weaker the prior, the smaller is λ and hence the closer the posterior mean is to the MLE.

- Can also show that the posterior mode is a convex combination of the prior mode and the MLE, and that it too converges to the MLE.

Posterior Variance

- The mean and mode are point estimates, but it is useful to know how much we can trust them. The variance of the posterior

$$p(\theta \mid \mathcal{D}) = \mathcal{Beta}\left(N_1 + a, N_0 + b\right)$$

is as follows:

$$\text{var}\left[\theta \mid \mathcal{D}\right] = \frac{\left(N_1 + a\right)\left(N_0 + b\right)}{\left(\alpha_0 + N\right)^2 \left(\alpha_0 + N + 1\right)}$$

$$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}$$
$$\theta \in [0, 1]$$

$$\text{E}(\theta) = \frac{\alpha}{\alpha+\beta}$$
$$\text{var}(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$
$$\text{mode}(\theta) = \frac{\alpha-1}{\alpha+\beta-2}$$

- We can simplify in the case that $N >> a, b$, to get

$$\text{var}[\theta|\mathcal{D}] \approx \frac{N_1 N_0}{NNN} = \frac{\hat{\theta}_{MLE}\left(1 - \hat{\theta}_{MLE}\right)}{N} = \frac{\hat{\theta}\left(1 - \hat{\theta}\right)}{N}$$

where $\hat{\theta} = \hat{\theta}_{MLE}$. Thus the error bar in our estimate is given by

$$\sigma \approx \sqrt{\frac{\hat{\theta}\left(1 - \hat{\theta}\right)}{N}}$$

Posterior Variance

$$\sigma \approx \sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{N}}$$

- We see that the uncertainty goes down at a rate of $1/\sqrt{N}$.
- Note, however, that the uncertainty is maximized when $\hat{\theta} = 0.5$, and is minimized when $\hat{\theta}$ is close to 0 or 1.
- This means it is easier to be sure that a coin is biased than to be sure that it is fair!

Posterior Predictive Distribution

- Consider *predicting the probability of heads in a single future trial*. We obtain the posterior mean:

$$p(\tilde{x}=1|\mathcal{D}) = \int_0^1 \theta \mathcal{B}e\mathcal{t}a(N_1+a, N_0+b)d\theta = \mathbb{E}[\theta|N_1+a, N_0+b] = \frac{N_1+a}{N+\alpha_0}$$

- For example, under uniform prior ($a=b=1$), this probability is:

$$p(\tilde{x}=1|\mathcal{D}) = \frac{N_1+1}{N+2}$$

- Add-One Smoothing: This justifies the common practice of adding 1 to the empirical counts, normalizing and then plugging them in.

- *How about if we plug in the MAP estimate? We can see that we don't then get this smoothing effect:*

$$p(\tilde{x}=1|\mathcal{D}) = \mathrm{mode}[\theta|N_1+a, N_0+b] = \frac{N_1+a-1}{N+\alpha_0-2} = \frac{N_1}{N} = \hat{\theta}_{MLE}$$

Posterior Predictive Distribution

- Consider predicting the probability of heads in a single future trial by plugging the MLE estimate as:

$$p(\tilde{x} = 1|\mathcal{D}) \approx Bern\left(\tilde{x} = 1|\hat{\theta}_{MLE}\right) = \hat{\theta}_{MLE} = \frac{N_1}{N}$$

- This often can be a troubling estimation (especially when we have few data available).

  - ✓ For example, if $N_1 = 0$ in $N = 3$ trials, then this predicts that the chance for getting heads on the next trial is zero!

$$p(\tilde{x} = 1|\mathcal{D}) \approx \hat{\theta}_{MLE} = \frac{0}{3} = 0$$

The Black Swan Paradox

$$p(\tilde{x} = 1|\mathcal{D} = \{TTT\}) \approx \hat{\theta}_{MLE} = \frac{0}{3} = 0$$

- This is called the zero count problem or the *sparse data problem*, and frequently occurs when estimating counts from small data sets.
- The zero-count problem is analogous to a problem in philosophy called *the black swan paradox.*
- This paradox was used to illustrate *the problem of induction* - drawing conclusions about the future from specific observations from the past.

- Taleb, N, (2007). *The Black Swan: The Impact of the Highly Improbable, 2nd Edt.* Random House.

Predicting the Outcome of Multiple Future Trials

- Suppose now we are interested in predicting the number of heads, x, in M future trials. For simplicity, we denote our posterior as $\mathcal{Beta}(a,b)$

$$p(x \mid \mathcal{D}, M) = \int_0^1 \mathcal{Bin}(x \mid \theta, M)\,\mathcal{Beta}(a,b)\,d\theta = \int_0^1 \binom{M}{x} \theta^x (1-\theta)^{M-x} \frac{\theta^{a-1}\theta^{b-1}}{Beta(a,b)}\,d\theta$$

$$= \binom{M}{x} \int_0^1 \frac{\theta^{a+x-1}\theta^{b+M-x-1}}{Beta(a,b)}\,d\theta = \binom{M}{x} \frac{Beta(a+x, b+M-x)}{Beta(a,b)}$$

- This is known as the *(compound) beta-binomial distribution*.

Beta-binomial

$$\theta \sim \text{Beta-bin}(n, \alpha, \beta)$$
$$p(\theta) = \text{Beta-bin}(\theta \mid n, \alpha, \beta)$$

'sample size'
n (positive integer)
'prior sample sizes'
$\alpha > 0, \beta > 0$

$$p(\theta) = \frac{\Gamma(n+1)}{\Gamma(\theta+1)\Gamma(n-\theta+1)} \frac{\Gamma(\alpha+\theta)\Gamma(n+\beta-\theta)}{\Gamma(\alpha+\beta+n)} \times \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}, \quad \theta = 0, 1, 2, \ldots, n$$

- This distribution has the following mean and variance

$$\mathbb{E}[x] = M \frac{a}{a+b}$$

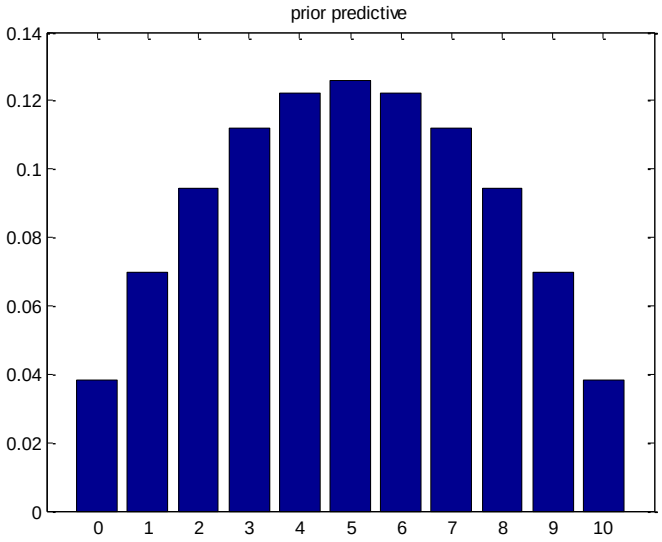
$$\text{var}[x] = M \frac{ab(a+b+M)}{(a+b)^2(a+b+1)}$$

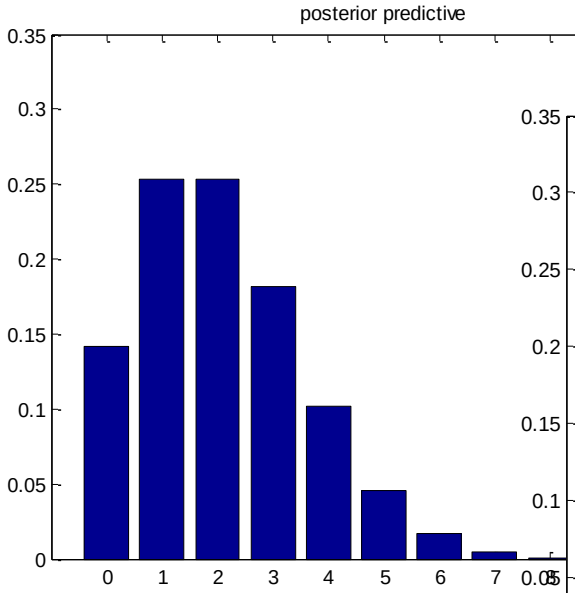
Note: here $x \leftarrow \theta, M \leftarrow n$

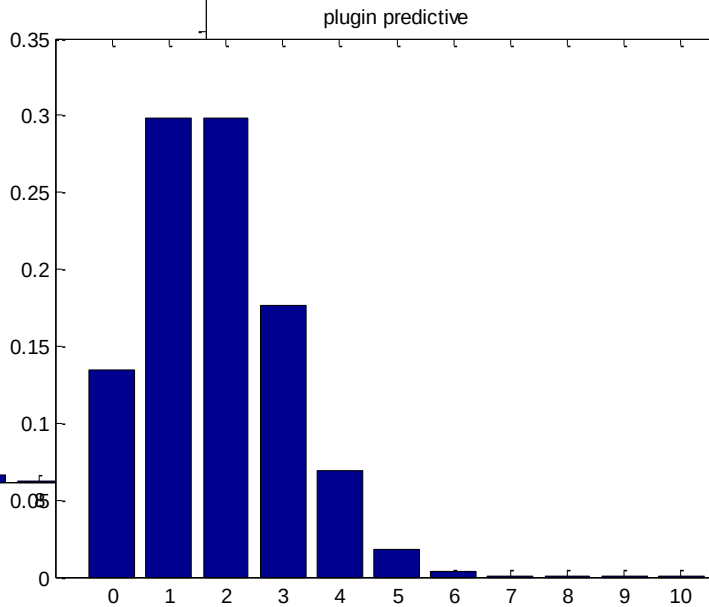
$$\mathrm{E}(\theta) = n\frac{\alpha}{\alpha+\beta}$$
$$\mathrm{var}(\theta) = n\frac{\alpha\beta(\alpha+\beta+n)}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Predicting the Outcome of Multiple Future Trials

- Posterior predictive distributions after seeing $N_1 = 3, N_0 = 17$.
- Note the *Bayesian prediction has longer tails, spreading its probablity mass more widely, and thus is less prone to overfitting and blackswan type paradoxes*.







Prior : $\mathcal{Beta}(2,2)$, `Prior' Predictive after : $N_1 = 3, N_0 = 17$

Posterior Predictive

Posterior Predictive Plug-in : $p(x|\mathcal{D}, M) = \mathcal{B}in(x|\hat{\theta}_{MAP}, M) = \binom{M}{x} \hat{\theta}_{MAP}{}^{x} (1 - \hat{\theta}_{MAP})^{M-x}$

Run *betaBinomPostPredDemo* from PMTK

The Dirichlet-Multinomial Model

- Suppose we observe N dice rolls $\mathcal{D} = \{x_1, x_2, ..., x_N\}$ where $x_i \in \{1, 2, ..., K\}$

- If we assume the data is iid, the likelihood has the form

$$p(\mathcal{D} \mid \boldsymbol{\theta}) \propto \prod_{k=1}^{K} \theta_k^{N_k}, \; N_k = \sum_{i=1}^{N} \mathbb{I}(x_i = k)$$

- N_k is the number of times event k occurred (*sufficient statistics for this model*).

- The likelihood for the multinomial model has the same form, up to an irrelevant constant factor.

The Dirichlet-Multinomial Model

- Since the parameter vector lives in the K –dimensional probability simplex, we need a prior that has support over this simplex. Ideally it would also be conjugate.

- The Dirichlet distribution satisfies both criteria.

- So we will use the following prior:

$$\mathcal{Dir}(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) = \frac{1}{Beta(\boldsymbol{\alpha})} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

$\theta \sim \text{Dirichlet}(\alpha_1, .., \alpha_k)$
$p(\theta) = \text{Dirichlet}(\theta | \alpha_1, .., \alpha_k)$

'prior sample sizes'
$\alpha_j > 0;\ \alpha_0 \equiv \sum_{j=1}^{k} \alpha_j$

$$p(\theta) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_k)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1}$$
$$\theta_1, .., \theta_k \geq 0; \sum_{j=1}^{k} \theta_j = 1$$

$$\mathrm{E}(\theta_j) = \frac{\alpha_j}{\alpha_0}$$
$$\mathrm{var}(\theta_j) = \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_0 + 1)}$$
$$\mathrm{cov}(\theta_i, \theta_j) = -\frac{\alpha_i \alpha_j}{\alpha_0^2(\alpha_0 + 1)}$$
$$\mathrm{mode}(\theta_j) = \frac{\alpha_j - 1}{\alpha_0 - k}$$

The Dirichlet-Multinomial Model Posterior

- Multiplying the likelihood by the prior, we find that the posterior is also Dirichlet:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \propto \prod_{k=1}^{K} \theta_k^{N_k} \theta_k^{\alpha_k - 1} = \mathcal{D}ir\left(\boldsymbol{\theta} \mid N_1 + \alpha_1, \ldots, N_K + \alpha_K\right)$$

- In deriving the mode of this posterior (i.e., the MAP estimate), we must enforce the constraint

$$\sum_{k=1}^{K} \theta_k = 1$$

- We can do this by using a Lagrange multiplier. The constrained objective function is given by the log likelihood plus log prior plus the constraint:

$$\ell(\boldsymbol{\theta}, \lambda) = \sum_{k=1}^{K} N_k \log \theta_k + \sum_{k=1}^{K} (\alpha_k - 1) \log \theta_k + \lambda \left(1 - \sum_{k=1}^{K} \theta_k\right)$$

The Dirichlet-Multinomial Model Posterior

$$\ell(\boldsymbol{\theta},\lambda)=\sum_{k=1}^{K}N_k\log\boldsymbol{\theta}_k+\sum_{k=1}^{K}(\alpha_k-1)\log\boldsymbol{\theta}_k+\lambda\left(1-\sum_{k=1}^{K}\boldsymbol{\theta}_k\right)$$

- To simplify notation, we define $N_k^{'}=N_k+\alpha_k-1$
- Taking derivatives with respect to λ yields: $\dfrac{\partial\ell(\boldsymbol{\theta},\lambda)}{\partial\lambda}=\left(1-\sum_{k=1}^{K}\boldsymbol{\theta}_k\right)=0$
- Taking derivatives with respect to $\boldsymbol{\theta}_k$ yields

$$\frac{\partial\ell(\boldsymbol{\theta},\lambda)}{\partial\boldsymbol{\theta}_k}=\frac{N_k^{'}}{\boldsymbol{\theta}_k}-\lambda=0$$

- We can solve for λ using the sum-to-one constraint:

$$\sum_{k=1}^{K}N_k^{'}=\lambda\sum_{k=1}^{K}\boldsymbol{\theta}_k\Rightarrow\lambda=\sum_{k=1}^{K}\left(N_k+\alpha_k-1\right)\Rightarrow\lambda=N+\sum_{k=1}^{K}\alpha_k-K$$

1

$$\lambda=N+\alpha_0-K,\ where:\alpha_0=\sum_{k=1}^{K}\alpha_k\ is\ the\ equivalent\ sample\ size\ of\ the\ prior$$

The Dirichlet-Multinomial Model Posterior

- Using $\frac{\partial \ell(\boldsymbol{\theta}, \lambda)}{\partial \boldsymbol{\theta}_k} = \frac{N_k'}{\boldsymbol{\theta}_k} - \lambda = 0$, the MAP estimate is given by

$$\hat{\theta}_k = \frac{N_k + \alpha_k - 1}{N + \alpha_0 - K}$$

- Compare this with the mode of the Dirichlet distribution $\mathcal{D}ir(\boldsymbol{\theta} \mid \alpha_1, \ldots, \alpha_K)$

$$\mathbb{E}[x_k] = \frac{\alpha_k}{\alpha_0}, \text{mode}[x_k] = \frac{\alpha_k - 1}{\alpha_0 - 1}, \text{var}[x_k] = \frac{\alpha_k(\alpha_0 - \alpha_k)}{\alpha_0^2(\alpha_0 + 1)}, \text{where}: \alpha_0 = \sum_{k=1}^{K} \alpha_k$$

- If we use a uniform prior, $a_k = 1$, we recover the MLE:

$$\hat{\theta}_k = \frac{N_k}{N}$$

- This is just the empirical fraction of times face k shows up.

Posterior Predictive

- The posterior predictive distribution for a single multinoulli trial is given by the following expression:

$$p(X=j \mid \mathcal{D}) = \int p(X=j \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta} = \int p(X=j \mid \theta_j) \int p(\boldsymbol{\theta}_{-j}, \theta_j \mid \mathcal{D}) d\boldsymbol{\theta}_{-j} d\theta_j$$

$$= \int \theta_j \, p(\theta_j \mid \mathcal{D}) d\theta_j = \mathbb{E}\left[\theta_j \mid \mathcal{D}\right]$$

$$= \frac{N_j + \alpha_j}{\sum_{k=1}^{K} \left(N_k + \alpha_k\right)} = \frac{N_j + \alpha_j}{N + \alpha_0}$$

$\theta \sim \text{Dirichlet}(\alpha_1, .., \alpha_k)$
$p(\theta) - \text{Dirichlet}(\theta \mid \alpha_1, .., \alpha_k)$

'prior sample sizes'
$\alpha_j > 0, \; \alpha_0 \equiv \sum_{j=1}^{k} \alpha_j$

- The above expression avoids the zero-count problem.

$$p(\theta) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_k)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1}$$
$$\theta_1, .., \theta_k \geq 0 : \sum_{j=1}^{k} \theta_j = 1$$

- *This form of Bayesian smoothing is even more important in the multinomial case than the binary case, since data sparsity increases once we start partitioning the data into many categories.*

$$\mathrm{E}(\theta_j) = \frac{\alpha_j}{\alpha_0}$$
$$\mathrm{var}(\theta_j) = \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_0 + 1)}$$
$$\mathrm{cov}(\theta_i, \theta_j) = -\frac{\alpha_i \alpha_j}{\alpha_0^2(\alpha_0 + 1)}$$
$$\mathrm{mode}(\theta_j) = \frac{\alpha_j - 1}{\alpha_0 - k}$$

Language Models Using Bag of Words

- Language modeling means predicting *which words might occur next in a sequence*. Assume that the i –th word, $X_i \in \{1, 2, ..., K\}$ is sampled independently from all the other words using a $\mathcal{C}at(\theta)$ distribution. This is called the *bag of words model*.

- Suppose we observe the following two sequences
  - *Mary had a little lamb, little lamb, little lamb,*
  - *Mary had a little lamb, its fleece was white snow*

- Furthermore, suppose our vocabulary consists of the following words:

Mary	lamb	little	big	fleece	white	black	snow	rain	unk
1	2	3	4	5	6	7	8	9	10

- Replace each word by its index into the vocabulary:

1 10 3 2 3 2 3 2
1 10 3 2 10 5 10 6 8

Token:	1	2	3	4	5	6	7	8	9	10
Word:	Mary	lamb	little	big	fleece	white	black	snow	rain	unk
Count:	2	4	4	0	1	1	0	1	0	4

Language Models Using Bag of Words

- Denote the above counts by N_j. If we use a $\mathcal{D}ir(a)$ prior for $\boldsymbol{\theta}$, the posterior predictive is just

$$p(\tilde{X} = j|\mathcal{D}) = \mathbb{E}(\theta_j \ |\mathcal{D}) = \frac{N_j + \alpha_j}{N + \alpha_0}$$

- If we set $a_j = 1$, we get $p(\tilde{X} = j|\mathcal{D}) = \mathbb{E}(\theta_j \ |\mathcal{D}) = \frac{N_j + 1}{17 + 10}$ from which:

$$p(\tilde{X} = j|\mathcal{D}) = \left\{\frac{3}{27}, \frac{5}{27}, \frac{5}{27}, \frac{1}{27}, \frac{2}{27}, \frac{2}{27}, \frac{1}{27}, \frac{2}{27}, \frac{1}{27}, \frac{5}{27}\right\}$$

- The modes of the predictive distribution are $X = 2$ ("lamb"), $X = 3$ ("little"), and $X = 10$ ("unk").

- *Note that the words ``big", ``black" and ``rain" are predicted to occur with non-zero probability in the future, even though they have never been seen before!*

Bayesian Analysis of the Uniform Distribution

- Consider $\mathcal{U}nif(0,\theta)$. The MLE is $\hat{\theta} = max(\mathcal{D})$. This is unsuitable for predicting future data since it puts zero probability mass outside the training data.
- We will perform a Bayesian analysis of the uniform distribution. The conjugate prior is the Pareto distribution,

$$p(\theta) = \mathcal{P}areto(\theta|b,K) = Kb^K\theta^{-(K+1)}\mathbb{I}(\theta \geq b), \text{ Mode}= b, \text{ Mean}= \begin{cases} \infty \; if \; K \leq 1 \\ \frac{Kb}{K-1} \; if \; K > 1 \end{cases}$$

- Given a Pareto prior, the joint distribution of θ and $\mathcal{D} = (x_1, \ldots, x_N)$ is:

$$p(\mathcal{D},\theta) = Kb^K\theta^{-(N+K+1)}\mathbb{I}\big(\theta \geq max(\mathcal{D},b)\big)$$

- Let $m = max(\mathcal{D})$. The evidence (probability that all N samples came from $\mathcal{U}nif(0,\theta)$) is

$$p(\mathcal{D}) = \int_{\max(m,b)}^{\infty} \frac{Kb^K}{\theta^{(N+K+1)}} d\theta = \begin{cases} \frac{K}{(N+K)b^N} \; if \; m \leq b \\ \frac{Kb^K}{(N+K)m^{(N+K)}} \; if \; m > b \end{cases}$$

Bayesian Analysis of the Uniform Distribution

$$p(\mathcal{D},\theta) = Kb^K\theta^{-(N+K+1)}\mathbb{I}\big(\theta \geq max(\mathcal{D},b)\big)$$

$$p(\mathcal{D}) = \begin{cases} \dfrac{K}{(N+K)b^N} & if\ m \leq b \\ \dfrac{Kb^K}{(N+K)m^{(N+K)}} & if\ m > b \end{cases}$$

- Using $c = \max(m,b)$, the posterior is given as:

$$p(\theta|\mathcal{D}) = \frac{p(\theta,\mathcal{D})}{p(\mathcal{D})} =$$

$$\begin{cases} Kb^K\theta^{-(N+K+1)}\dfrac{(N+K)b^N}{K} = \dfrac{(N+K)b^{(N+K)}}{\theta^{N+K+1}} & if\ m \leq b \leq \theta, \\ Kb^K\theta^{-(N+K+1)}\dfrac{(N+K)m^{(N+K)}}{Kb^K} = \dfrac{(N+K)m^{(N+K)}}{\theta^{N+K+1}} & if\ b < m \leq \theta \end{cases} =$$

$$\mathcal{P}areto(\theta|c,N+K)$$

Bayesian Analysis of the Uniform Distribution

$$p(\theta|\mathcal{D}) = \mathcal{P}areto(\theta|c, N+K),\ c = \max(m, b)$$

- With this posterior, the predictive distribution can be computed as follows:

$$p(x|\mathcal{D}) = \begin{cases} \int_c^\infty \frac{1}{\theta}(N+K)c^{(N+K)}\theta^{-(N+K+1)}d\theta = \frac{(N+K)}{(N+K+1)c}\ if\ x \le c \\ \int_x^\infty \frac{1}{\theta}(N+K)c^{(N+K)}\theta^{-(N+K+1)}d\theta = \frac{(N+K)c^{(N+K)}}{(N+K+1)x^{(N+K+1)}}\ if\ x > c \end{cases}$$

- With a non-informatice prior, $b = K = 0$, and if $\mathcal{D} = \{100\}$, so that $m = 100, N = 1$, then:

$$p(x|\mathcal{D}) = \begin{cases} \frac{1}{2m}\ if\ x \le c = m = 100 \\ \frac{m}{2x^2}\ if\ x > c \end{cases}$$

- For implementation see ParetoDemoTaxicab, from PMTK

Naive Bayes Classifier

- We discuss how to classify vectors of discrete-valued features, $\boldsymbol{x} \in \{1, 2, ..., K\}^D$ where K is the number of values for each feature, and D *is the number of features*.

- The simplest approach is to *assume the features are conditionally independent given the class label.* This allows us to write the class conditional density as a product of one dimensional densities:Type equation here.

$$p(\boldsymbol{x}|y = c, \boldsymbol{\theta}) = \prod_{j=1}^{D} p\left(x_j|y = c, \boldsymbol{\theta}_{jc}\right)$$

- The resulting model is called *a naive Bayes classifier (NBC).*

- It is called *"naive" since in practice the features x_j are not independent, even conditional on the class label c.*

Naive Bayes Classifiers

$$p(\boldsymbol{x}|y=c,\boldsymbol{\theta}) = \prod_{j=1}^{D} p\left(x_j|y=c,\boldsymbol{\theta}_{jc}\right)$$

- The naive Bayes assumption as unreasonable as it might be, it often results in classifiers that work well.
  - The reason for this is that *the model* is simple (it only has $\mathcal{O}(CD)$ parameters, for C classes and D features)
  - Hence the model is *relatively immune to overfitting.*

- Domingos, P. and M. Pazzani (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning 29*, 103– 130.

Naive Bayes Classifiers: Examples

- The form of the class-conditional density depends on the type of each feature. Three examples are shown below:

  - For *real-valued features,* we can use the Gaussian distribution:

$$p(\boldsymbol{x}|y=c,\theta)=\prod_{j=1}^{D}\mathcal{N}\left(x_j|\mu_{jc},\sigma_{jc}^2\right)$$

    where μ_{jc} is the mean of feature j in objects of class c, and σ_{jc}^2 is its variance.

  - For *binary features*, $x_j \in \{0,1\}$, we can use the Bernoulli distribution

$$p(\boldsymbol{x}|y=c,\theta)=\prod_{j=1}^{D}\mathcal{Ber}\left(x_j|\mu_{jc}\right)$$

    where μ_{jc} is the probability that feature j occurs in class c. This is called the *multivariate Bernoulli naive Bayes model.*

Naïve Bayes Classifiers: Examples

- The form of the class-conditional density depends on the type of each feature.

  - For *categorical features*, $x_j \in \{1, 2, ..., K\}$, we model using the *multinoulli (Categorical) distribution* (this is the same as the multinomial distribution with $N = 1$)

$$p(\boldsymbol{x}|y = c, \theta) = \prod_{j=1}^{D} \mathcal{C}at(x_j|\boldsymbol{\mu}_{jc})$$

    where $\boldsymbol{\mu}_{jc}$ is a vector of the probabilities over the K possible values for x_j in class c.

- *Training a naive Bayes classifier:*

  - This usually refers to *computing the MLE and MAP estimates for the model parameters*

MLE for Naïve Bayes Classifier

- The *probability for a single data case (known features & class label)* is

$$p(\boldsymbol{x}_i, y_i \mid \boldsymbol{\theta}) = p(y_i \mid \pi) \prod_{j=1}^{D} p(x_{ij} \mid y_i, \boldsymbol{\theta}_j) = \left(\prod_c \pi_c^{\mathbb{I}(y_i=c)} \right) \prod_j \prod_c p(x_{ij} \mid \boldsymbol{\theta}_{jc})^{\mathbb{I}(y_i=c)}$$

- Hence the joint log-likelihood is given by

$$\log p(\mathcal{D} \mid \boldsymbol{\theta}) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log p(x_{ij} \mid \boldsymbol{\theta}_{jc})$$

- The likelihood decomposes into one term concerning π and $D \times C$ terms containing the $\boldsymbol{\theta}_{jc}$. We can *optimize these parameters separately.*

- *The MLE for the class prior* is given (proof as given earlier) by

$$\hat{\pi}_c = \frac{N_c}{N}$$

where $N_c = \sum_i \mathbb{I}(y_i = c)$ *is the number of examples in class* c.

MLE for Naïve Bayes Classifier

$$\log p(\mathcal{D} \mid \boldsymbol{\theta}) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log p\left(x_{ij} \mid \boldsymbol{\theta}_{jc}\right)$$

- The MLE for $\boldsymbol{\theta}_{jc}$ depends on the type of distribution we use for each feature. For simplicity, let us *suppose all features are binary*, so

$$x_j \mid y = c \sim \mathcal{B}er(\boldsymbol{\theta}_{jc})$$

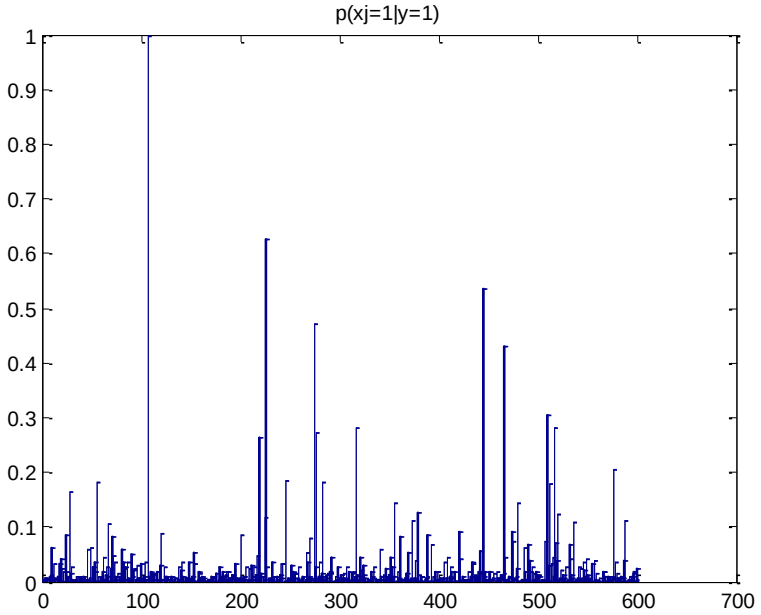
- In this case, it is easy to show from the MLE becomes

$$\hat{\theta}_{jc} = \frac{N_{jc}}{N_c},\ N_c = \sum_i \mathbb{I}(y_i = c),\ N_{jc} = \sum_{i:y_i=c} \mathbb{I}(x_{ij} = 1)$$

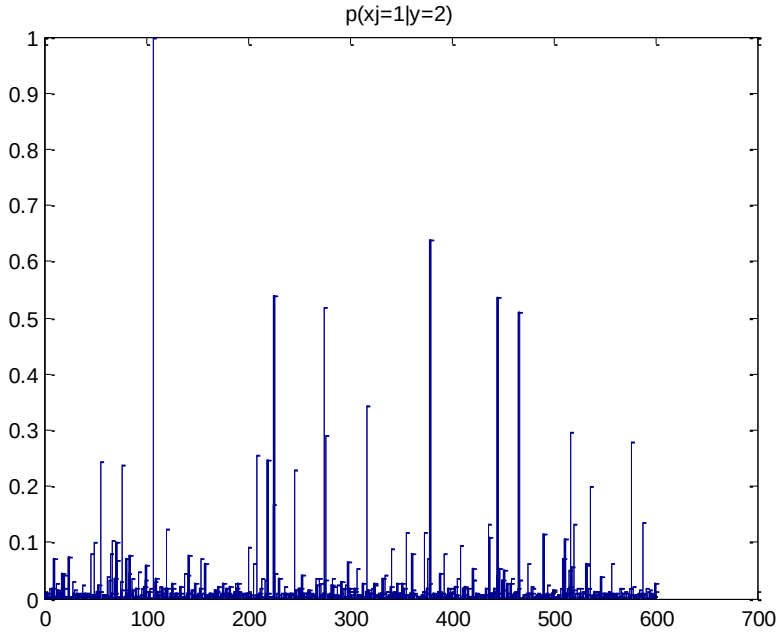
  - *N_c is the number of data in class c.*
  - *N_{jc} is the number of times in these N_c data that feature j is on.*

MLE for Naïve Bayes Classifier

- The figures below give an example with 2 *classes and* 600 *binary features, representing the presence or absence of words in a bag-of-words model.* The plot visualizes the $\boldsymbol{\theta}_c$ vectors for the two classes.
- The spike at location 107 corresponds to the word `subject', which occurs in both classes with probability 1 (this is overfitting)



Class conditional densities $p(x_j = 1|y = c)$ for two document classes, corresponding to "X windows" and "MS windows".



Run *naiveBayesBowDemo* from PMTK

MLE for Naïve Bayes Classifier: Algorithm

- We summarize below the MLE calculation algorithm for binary features.

```
1 N_c = 0, N_jc = 0;
2 for i = 1 : N do
3     c = y_i // Class label of i'th example;
4     N_c := N_c + 1 ;
5     for j = 1 : D do
6         if x_ij = 1 then
7             N_jc := N_jc + 1
8 π̂_c = N_c / N, θ̂_jc = N_jc / N_c
```

From PMTK

Bayesian Naïve Bayes Classifier

- The trouble with MLE (as shown earlier) is that it can overfit
- A Bayesian approach is needed. Use a factored conjugate prior:

$$p(\boldsymbol{\theta}) = p(\pi)\prod_{j=1}^{D}\prod_{c=1}^{C} p(\theta_{jc}) \propto \prod_{k=1}^{C} \pi_k^{\alpha_k - 1}\prod_{j=1}^{D}\prod_{c=1}^{C} \theta_{jc}^{\beta_0 - 1}\left(1-\theta_{jc}\right)^{\beta_1 - 1}$$

- We use a $\mathcal{D}ir(\boldsymbol{\alpha})$ prior for π and a $\mathcal{B}eta(\beta_0, \beta_1)$ prior for each θ_{jc}. *Often we take $\boldsymbol{\alpha} = \mathbf{1}$ and $\boldsymbol{\beta} = \mathbf{1}$, corresponding to add-one or Laplace smoothing.*
- Combining the factored likelihood (for binary features)

$$p\left(\mathcal{D} \mid \boldsymbol{\theta}\right) = \prod_{c} \pi_c^{N_c}\prod_{j}\prod_{c} \theta_{jc}^{N_{jc}}\left(1-\theta_{jc}\right)^{N_c - N_{jc}}$$

with the factored prior above gives the following factored posterior:

$$p(\theta \mid \mathcal{D}) = p(\pi \mid \mathcal{D})\prod_{j=1}^{D}\prod_{c=1}^{C} p(\theta_{jc} \mid \mathcal{D}) = \prod_{c}^{C} \pi_c^{N_c + \alpha_c - 1}\prod_{j=1}^{D}\prod_{c}^{C} \theta_{jc}^{N_{jc} + \beta_0 - 1}\left(1-\theta_{jc}\right)^{N_c - N_{jc} + \beta_1 - 1}$$

$$p(\pi \mid \mathcal{D}) = \mathcal{D}ir\left(N_1 + \alpha_1, \ldots, N_C + \alpha_C\right),\ p(\theta_{jc} \mid \mathcal{D}) = \mathcal{B}eta\left(N_{jc} + \beta_0, N_c - N_{jc} + \beta_1\right)$$

- *We thus simply update the prior counts with the empirical counts from the likelihood.*

Using the Model for Prediction

- At test time, the goal is to *compute the predictive distribution:*

$$p(y=c \mid \boldsymbol{x}, \mathcal{D}) \propto p(y=c \mid \mathcal{D}) \prod_{j=1}^{D} p(x_j \mid y=c, \mathcal{D})$$

- The correct Bayesian procedure is to integrate out the unknown parameters:

$$p(y=c \mid \boldsymbol{x}, \mathcal{D}) \propto \left[\int \mathcal{C}at(y=c \mid \pi) p(\pi \mid \mathcal{D}) d\pi \right] \prod_{j=1}^{D} \left[\int \mathcal{B}er(x_j \mid y=c, \theta_{jc}) p(\theta_{jc} \mid \mathcal{D}) d\theta_{jc} \right]$$

Using the Model for Prediction

- This is easy to do if the posterior is Dirichlet. In particular, recall that the predictive distribution gives the posterior mean of the parameters:

$$p(\tilde{X} = j|\mathcal{D}) = \mathbb{E}(\theta_j \ |\mathcal{D}) = \frac{N_j + \alpha_j}{N + \alpha_0}$$

- Thus we know *the posterior predictive density can be obtained by simply* ***plugging in the posterior mean parameters*** $\bar{\theta}$. Hence

$$p(y = c|\boldsymbol{x}, \mathcal{D}) \propto \bar{\pi}_c \prod_{j=1}^{D} \left(\bar{\theta}_{jc}\right)^{\mathbb{I}(x_j=1)} \left(1 - \bar{\theta}_{jc}\right)^{\mathbb{I}(x_j=0)}$$

$$\bar{\theta}_{jc} = \frac{N_{jc} + \beta_0}{N_c + \beta_0 + \beta_1} \quad \textit{posterior mean of Beta}$$

$$\bar{\pi}_c = \frac{N_c + \alpha_c}{N + \alpha_0}, \ \alpha_0 = \sum_c \alpha_c \quad \textit{posterior mean of Dirichlet}$$

Plugin Approximation

- If we approximate the posterior by a single point, $p(\theta|D) \approx \delta_{\hat{\theta}}(\theta)$ where θ is the MLE or MAP, then the posterior predictive density is obtained by plugging in the parameters to yield a virtually identical rule.

$$p(y = c|\boldsymbol{x}, \mathcal{D}) \propto \hat{\pi}_c \prod_{j=1}^{D} \left(\hat{\theta}_{jc}\right)^{\mathbb{I}(x_j=1)} \left(1 - \hat{\theta}_{jc}\right)^{\mathbb{I}(x_j=0)}$$

- The only difference is we replaced the posterior mean $\bar{\theta}$ with the MAP or MLE $\hat{\theta}$.

- This small difference is important since *the posterior mean results in less overfitting.*

The Log-Sum-Exp Trick

- A naive implementation of

$$p(y=c \mid \boldsymbol{x}, \boldsymbol{\theta}) \propto \frac{p(\boldsymbol{x} \mid y=c, \boldsymbol{\theta})\, p(y=c \mid \boldsymbol{\theta})}{\sum_{c'} p(\boldsymbol{x} \mid y=c', \boldsymbol{\theta})\, p(y=c' \mid \boldsymbol{\theta})}$$

can fail due to *numerical underflow*. The problem is that $p(\boldsymbol{x} \mid y = c)$ *is often a very small number, especially if* $\boldsymbol{x}$ *is a high-dimensional vector.*

- This is because we require that $\sum_{\boldsymbol{x}} p(\boldsymbol{x} \mid y) = 1$, so *the probability of observing any particular high-dimensional vector is* $\boldsymbol{x}$ *small*.

- The obvious solution is to *take logs when applying Bayes rule*, as follows:

$$\log p(y=c \mid \boldsymbol{x}) = b_c - \log\left[\sum_{c'=1}^{C} e^{b_{c'}}\right], b_c = \log p(\boldsymbol{x} \mid y=c) + \log p(y=c)$$

The Log-Sum-Exp Trick

- However, this requires evaluating the following expression

$$\log\left[\sum_{c'=1}^{C} e^{b_{c'}}\right] = \log \sum_{c'} p(y = c', \boldsymbol{x}) = \log p(\boldsymbol{x})$$

and we cannot add up in the log domain.

- One can *factor out the largest term and represent the remaining numbers relative to that*, e.g.

$$\log\left(e^{-120} + e^{-121}\right) = \log\left(e^{0} + e^{-1}\right) - 120$$

- In general, we have

$$\log\left[\sum_c e^{b_c}\right] = \log\left[\left(\sum_c e^{b_c - B}\right) e^{B}\right] = \log\left(\sum_c e^{b_c - B}\right) + B$$

- This l*og-sum-exp trick* is widely used.

Run *naiveBayesPredict* from PMTK

Predictions with the Naïve Bayes Classifier: Algorithm

- We summarize below the prediction NBC Algorithm for binary features.

for $i = 1 : N$ **do**
  for $c = 1 : C$ **do**
    $L_{ic} = \log \hat{\pi}_c$;
    for $j = 1 : D$ **do**
      if $x_{ij} = 1$ **then** $L_{ic} := L_{ic} + \log \hat{\theta}_{jc}$ **else** $L_{ic} := L_{ic} + \log(1 - \hat{\theta}_{jc})$
  $p_{ic} = \exp(L_{ic} - \text{logsumexp}(L_{i,:}))$;
  $\hat{y}_i = \text{argmax}_c \, p_{ic}$;

From PMTK

$$\log p(y = c \mid \boldsymbol{x}) = b_c - \log\left[\sum_{c'=1}^{C} e^{b_{c'}}\right], b_c = \log p(\boldsymbol{x} \mid y = c) + \log p(y = c)$$

Feature Selection Using Mutual Information

- Since an NBC is fitting a joint distribution over many features
  - (a) it can suffer from overfitting and
  - (b) the run-time cost is $\mathcal{O}(D)$ which may be too high

- We tackle these problems by performing *feature selection* i.e. by *removing irrelevant features*.
  - Evaluate the relevance of each feature separately, and then take the top K, where K is chosen based on some *tradeoff between accuracy and complexity*.
  - This approach is known as *variable ranking, filtering, or screening*.

- One way to measure relevance is to *use mutual information between feature X_j and the class label Y*:

$$I(X,Y) = \sum_{x_j} \sum_{y} p(x_j, y) \log \frac{p(x_j, y)}{p(x_j)\, p(y)}$$

Feature Selection Using Mutual Information

- *The mutual information can be thought of as the reduction in entropy on the label distribution once we observe the value of feature j.*

- *If the features are binary*, it is easy to show that the MI can be computed as follows

$$I_j = \sum_c \left[\theta_{jc}\pi_c \log \frac{\theta_{jc}}{\theta_j} + (1-\theta_{jc})\pi_c \log \frac{1-\theta_{jc}}{1-\theta_j} \right]$$

where $\pi_c = p(y = c), \theta_{jc} = p(x_j = 1|y = c)$, and $\theta_j = p(x_j = 1) = \sum_c \pi_c \theta_{jc}$

$$I_j = \sum_{x=0,1} \sum_c p(x_j = x \mid y)\, p(y = c) \log \frac{p(x_j = x \mid y = c)}{p(x_j = x)}$$

$$= \sum_c p(x_j = 1 \mid y)\, p(y = c) \log \frac{p(x_j = 1 \mid y = c)}{p(x_j = 1)} + \sum_c p(x_j = 0 \mid y = c)\, p(y = c) \log \frac{p(x_j = 0 \mid y = c)}{p(x_j = 0)}$$

$$= \sum_c \left[\theta_{jc}\pi_c \log \frac{\theta_{jc}}{\theta_j} + (1-\theta_{jc})\pi_c \log \frac{1-\theta_{jc}}{1-\theta_j} \right]$$

- *All of these quantities are computed when fitting the naive Bayes classifier.*

Feature Selection Using Mutual Information

- *The words with highest MI are much more discriminative than the words which are most probable.*
- Most Probable Words: In the earlier example, the most probable word in both classes is "subject", which always occurs because this is newsgroup data which always has a subject line. Obviously this is not very discriminative.
- Most Discriminative Words: The words with highest MI with the class label are (in decreasing order) "windows", "microsoft", "DOS" and "motif". This makes sense, since the two classes correspond to Microsoft Windows and X Windows.

class 1	prob	class 2	prob	highest MI	MI
subject	0.998	subject	0.998	windows	0.215
this	0.628	windows	0.639	microsoft	0.095
with	0.535	this	0.540	dos	0.092
but	0.471	with	0.538	motif	0.078
you	0.431	but	0.518	window	0.067

Run *naiveBayesBowDemo* from PMTK

Classifying Documents Using Bag of Words

- Consider classifying text documents into different categories.
- Represent each document as a binary vector recording whether each word is present or not, so $x_{ij} = 1$ *iff word* j *occurs in document* i, *otherwise* $x_{ij} = 0$. Then use the following class conditional density:

$$p(\boldsymbol{x}_i \mid y_i = c, \boldsymbol{\theta}) = \prod_{j=1}^{D} \mathcal{B}er(x_{ij} \mid \theta_{jc}) = \prod_{j=1}^{D} \theta_{jc}^{\mathbb{I}(x_{ij})} (1-\theta_{jc})^{\mathbb{I}(1-x_{ij})}$$

- This is called the ***Bernoulli product model***, or the ***binary independence model***.
- *Note that ignoring the number of times each word occurs in a document (as is the case with the model above) loses some information*.
- For more accurate representation we need to *count the number of occurrences of each word.*

- McCallum, A. and K. Nigam (1998). A comparison of event models for naive Bayes text classification. In *AAAI/ICML workshop on Learning for Text Categorization*.

Classifying Documents Using Bag of Words

- Specifically, let $\boldsymbol{x}_i$ be a *vector of counts for document i*, so $x_{ij} \in \{0,1 \ldots., N_i\}$, where N_i is the number of terms in document i

$$\sum_{j=1}^{D} x_{ij} = N_i$$

Because of this constraint, the features are not independent

- *For the class conditional densities, we can use a multinomial distribution*:

$$p(\boldsymbol{x}_i \mid y_i = c, \boldsymbol{\theta}) = \mathcal{M}\boldsymbol{u}(\boldsymbol{x}_i \mid N_i, \boldsymbol{\theta}_c) = \frac{N_i!}{\prod_{j=1}^{D} x_{ij}!} \prod_{j=1}^{D} \theta_{jc}^{x_{ij}}$$

- Here we *assume that the document length N_i is independent of the class.*

- θ_{jc} is the probability of generating word j in documents of class c; these parameters satisfy the constraint that

for each class c. $$\sum_{j=1}^{D} \theta_{jc} = 1$$

Burstiness of Words

- The multinomial classifier is easy to train and use for predictions, however it *does not take into account the burstiness of word usage*.

- Words occur in bursts: *most words never appear in any given document, but if they do appear once, they are likely to appear more than once.*

- The multinomial model cannot capture the burstiness phenomenon. To see why, note that the equation

$$p(\mathbf{x}_i \mid y_i = c, \boldsymbol{\theta}) = \mathcal{M}\boldsymbol{u}(\mathbf{x}_i \mid N_i, \boldsymbol{\theta}_c) = \frac{N_i!}{\prod_{j=1}^{D} x_{ij}!} \prod_{j=1}^{D} \theta_{jc}^{x_{ij}}$$

has the form $\theta_{jc}^{N_{ij}}$ and *since* $\theta_{jc} << 1$ *for rare words, it becomes increasingly unlikely to generate many of them.*

Multinomial Document Classifiers

- For more frequent words, the decay rate is not as fast as $\theta_{jc}^{N_{ij}}$. To see why intuitively, note that the most frequent words are function words which are not specific to the class, such as “and”, “the”, and “but”.
- *The independence assumption is more reasonable for common words: e.g.* the chance of the word “and” occurring is pretty much the same no matter how many times it has previously occurred.
- Since rare words are the ones that matter most for classification purposes, these are the ones we want to model the most carefully.
- Various ad hoc heuristics have been proposed to improve the performance of the multinomial document classifier.

- Rennie, J., L. Shih, J. Teevan, and D. Karger (2003). Tackling the poor assumptions of naive Bayes text classifiers. In *Intl. Conf. on Machine Learning*.
- Madsen, R., D. Kauchak, and C. Elkan (2005). Modeling word burstiness using the Dirichlet distribution. In *Intl. Conf. on Machine Learning*.

Compound Multinomial (DCM) Density

- Suppose we simply replace the multinomial class conditional density with the *Dirichlet Compound Multinomial (DCM) density,* defined as:

$$p(\boldsymbol{x}_i \mid y_i = c, \boldsymbol{\alpha}) = \int \mathcal{M}\boldsymbol{u}(\boldsymbol{x}_i \mid N_i, \boldsymbol{\theta}_c)\mathcal{D}\boldsymbol{i}\boldsymbol{r}(\boldsymbol{\theta}_c \mid \boldsymbol{\alpha}_c)d\boldsymbol{\theta}_c = \frac{N_i!}{\prod_{j=1}^{D} x_{ij}!}\frac{B(\boldsymbol{x}_i+\boldsymbol{\alpha}_c)}{B(\boldsymbol{\alpha}_c)} \qquad B(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^{K}\Gamma(\alpha_k)}{\Gamma\left(\sum_k \alpha_k\right)}$$

- This DCM density captures the burstiness phenomenon.
- *After seeing one occurrence of a word, say word j, the posterior counts on θ_j gets updated, making another occurrence of word j more likely.* By contrast, if θ_j is fixed, then the occurrences of each word are independent.
- The multinomial model corresponds to drawing a ball from an urn with K colors of ball, recording its color, and then replacing it. By contrast, *the DCM model corresponds to drawing a ball, recording its color, and then* ***replacing it with one additional copy*** *(Polya urn)*

Compound Multinomial (DCM) Density

- The DCM as the class conditional density gives much better results than using the multinomial.
- It has performance comparable to state of the art methods.
- Fitting the DCM model is however more complex.

- Minka, T. (2000). Estimating a Dirichlet distribution. Technical report, MIT.
- Elkan, C. (2006). Clustering documents with an exponential family approximation of the Dirichlet compound multinomial model. In *Intl. Conf. on Machine Learning*.
- Madsen, R., D. Kauchak, and C. Elkan (2005). Modeling word burstiness using the Dirichlet distribution. In *Intl. Conf. on Machine Learning*.